

**Sistemas Paralelos e Distribuídos**  
**Turnos Prático e Laboratorial**  
**Atividade 2 : Sistemas Distribuídos**

**Projeto de um sistema distribuído para gestão de “chave-valor”**

**1. Sumário**

O objetivo é fazer um projeto, que visa desenvolver e implementar um sistema de armazenamento de *key-value* distribuído. O sistema deve ser capaz de armazenar, recuperar e gerir pares *key-value* fazendo a coordenação de vários nós (num ambiente distribuído). O sistema deve também lidar com as principais questões de sistemas distribuídos, tais como tolerância a faltas, consistência, disponibilidade e escalabilidade.

**2. Requisitos mínimos a serem atendidos:**

- Funcionalidade mínima: PUT (key,value), GET (key) e DELETE (key).
- Comunicação entre cliente e servidor via REST API.
- Uso de *key-values* de qualquer tipo (tipo *string* pode ser usado nesse caso).
- Armazenamento persistente de valores.
- Uso de múltiplos nós com distribuição balanceada.
- O sistema deve ser escalável de forma automatizada.
- O sistema deve ser capaz de responder a falhas de forma automatizada.
- O sistema deve ser modularizado, utilizando o conceito de micro-serviços.
- A consistência dos dados deve ser endereçada e justificada.
- O sistema tem de ser portátil, com automatismo para instalação e *setup* instantâneo para *docker* em sistema GNU/Linux.
- O sistema deve permitir *health checks* e testes unitários para validação de estados operacionais.
- Tecnologias empregadas devem privilegiar o *open-source*.

**3. Documentação necessária:**

- Diagrama de arquitetura do sistema.
- Manual da API e uso do sistema (e.g.: *swagger* com exemplos de comandos).
- Descrição detalhada e passo a passo para instalação e uso em cloud e standalone.
- Especificações de limites e capacidades do sistema.
- Relatório de testes de carga com justificação de dimensões e limites do sistema.
- Bibliografia formal do projeto, incluindo indicação de uso de conteúdo produzido por auxílio de plataformas de IA.

#### **4. Qualidades pertinentes aos sistemas distribuídos:**

O projeto deverá conter explicação clara (em sua documentação) de como os seguintes aspectos de sistemas distribuídos são abordados na solução:

- Concorrência
- Escalabilidade
- Tolerância a faltas
- Consistência
- Coordenação de recursos compartilhados

#### **5. Produto final esperado**

O resultado final do projeto - objeto de avaliação - deve conter:

- Documentação: Desenho e implementação (conforme especificado no item 3)
- Manual de utilização
- Código fonte
- Relatórios de testes
- Demo do sistema

#### **6. Pontos a serem avaliados**

A avaliação do projeto respeitará os seguintes critérios:

- Qualidade de documentação (4 pontos)
- Qualidade do manual (4 pontos)
- Abordagem adequada aos principais quesitos de sistemas distribuídos (concorrência, escalabilidade, etc) (4 pontos)
- Funcionalidade correta (4 pontos)
- Qualidade de código (2 pontos)
- Performance do sistema (2 pontos)

## 7. Regras gerais:

- O projeto deve ser individual.
- Linguagem e protocolos são livres, com exceção de dois pontos: o servidor de interface pública deve usar REST API com HTTP e todos os serviços devem executar em *containers Linux*.
- O projeto deve ter um arquivo *bash* para bootstrap e *entrypoint* com o seguinte nome: **start.sh**. Isso não implica que o projeto tenha de ser todo em shell bash. Este é somente um *script* que inicia o setup e execução do sistema.
- As chamadas implementadas deverão seguir os seguintes formatos (parâmetros deverão utilizar os mesmos nomes, caso contrário o programa será considerado incorreto):
  - method **GET**, argument: **key**, deve retornar **json {"data": {"value":<foo>}}**
  - method **PUT**, argument request body **json {"data": {"key":<foo>,"value":<foo> }},** deve retornar **HTTP response code**.
  - method **DELETE**, argument: **key**, deve retornar **HTTP response code**.
- O projeto entregue deve incluir uma versão demo que permita a visualização e avaliação do sistema em suas funções principais.
- Todo o código e documentação devem ser armazenados e apresentados em um repositório no GitHub. O repositório deverá ser privado com acesso permitido ao criador e aos professores (*git handlers: margaridamadeira e rzuolo*).
- A conta de utilizador no Github deverá ser associado ao email do aluno no domínio ualg.pt, e o perfil do Github deverá obrigatoriamente corresponder ao número do aluno (e.g.: <https://github.com/a66264>)
- Toda documentação deverá ser apresentada num formato *Markup* (como típicos arquivos "Readme").
- A conclusão da atividade implica a disponibilização do produto final na plataforma Github.
- A entrega (registo do repositório na resposta da atividade-02 da tutoria) deverá ser realizada até dia 23 de maio de 2025, às 21:00 h e incluir, pelo menos, as peças textuais (Desenho e implementação - item 3 e Manual de utilização).